

Portfolio
*Synthetische Baummodelle aus iterierten
diskreten Graphen*
im Rahmen des Moduls M.Inf.1261

von Sabine Wolff

im Master of Education
Zertifikat Drittfach Informatik

betreut von Dr. Katarina Smoleňová und Prof. Dr. Winfried Kurth

Abgabedatum: 31.03.2013

WS 12/13
(3. Fachsemester)

Inhaltsverzeichnis

1	Einleitung	3
2	Bisherige Arbeiten	3
2.1	Selbstorganisierende und bildbasierte Baummodelle	4
2.1.1	Selbstorganisierende Baummodelle: „Self-organizing tree models for image synthesis“	4
2.1.2	Bildbasierte Baummodelle: „Approximate Image-Based Tree-Modeling using Particle Flows“	7
3	Der Algorithmus von Xu und Mould	7
3.1	Variation der Parameter	10
3.1.1	Form des initialen Graphvolumens	10
3.2	Erzeugen der Subgraphen	13
3.3	Lebensspanne	15
3.4	Umwelteinflüsse	16
3.5	Ergebnisse und Evaluation	17
4	Zusammenfassung und Ausblick	22
4.1	Diskussion	22

1 Einleitung

Bäume sind üblicher Bestandteil von Landschaften und somit häufig in virtuellen Welten wie Computerspielen und Filmen zu finden. Auch sie besitzen eine komplexe Struktur, die den Charakter ihres Aussehens ausmacht und die nur mit Mühe von Menschen nachgeahmt werden kann. Xu und Mould stellen in ihren Papern „Synthetic Tree Models from Iterated Discrete Graphs“ [1] und „A procedural method for irregular tree models“ [2] eine prozedurale Methode zum Generieren von Baummodellen vor, deren zugrundeliegender Algorithmus darauf beruht einen Pfad mit optimalen Kosten in einem unregelmäßigen Gitter zu suchen.

Nach einem kurzen Überblick über weitere Methoden zum Erzeugen von digitalen Baummodellen (vgl. Abschnitt 2), wird in diesem Bericht die Methode von Xu und Mould vorgestellt. Dabei wird zunächst das Grundprinzip erläutert (vgl. Abschnitt 3) und die Möglichkeiten auf resultierende Modelle Einfluss zu nehmen ausgeführt (vgl. Abschnitt 3.1). Anschließend werden die Ergebnisse vorgestellt und mit Modellen, die mit anderen Methoden erstellt wurden, verglichen (vgl. Abschnitt 3.5). Am Ende des Berichtes erfolgt eine kurze Diskussion der vorgestellten Methode (vgl. Abschnitt 4.1).

2 Bisherige Arbeiten

Da das Modellieren von Bäumen bereits eine lange Tradition in der Computergraphik hat, gibt es viele verschiedene Ansätze. Eine Herangehensweise ist die Verwendung von Lindenmayersystemen (L-Systeme), welche Grammatiken mit paralleler Anwendung von Ersetzungsregeln sind [3]. Diese wurden als theoretische Umgebung entwickelt, um z.B. Pflanzen zu modellieren. Bei der Methode wird verwendet, dass eine Pflanze aus verschiedenen Komponenten wie Zellen, Blättern, Ästen oder Knospen in verschiedenen Zuständen besteht, welche jeweils durch einen Buchstaben repräsentiert werden. Eine Aneinanderreihung dieser Buchstaben formt ein Wort, das die gesamte Pflanze repräsentiert. Durch die parallele Anwendung von Produktionsregeln auf Buchstaben des Wortes ergibt sich eine Simulation des biologischen Wachstums, welches bei einer Pflanze auch an verschiedenen Stellen gleichzeitig geschieht [3].

Eine Schwierigkeit bei grammatikbasierten Modellierungsmethoden ist, die Abfolge der angewendeten Produktionsregeln so zu kontrollieren, dass das erhaltene Produkt vordefinierten Vorgaben und Formen entspricht. Die Arbeit von Talton et al. [4] beschreibt mit Metropolis Procedural Modelling eine dafür geeignete Möglichkeit, die jedoch eine hohe Anzahl von Stichprobennahmen benötigt, was sehr zeitaufwändig sein kann. Einen biologisch motivierten Kontrollmechanismus für die Modellierung von Bäumen zeigt Runions et al. [5] in Form der Raumausnutzung, bei der eine Erweiterung des Baumes nur dann ausgeführt wird, wenn sich die neuen Pflanzenkomponenten nicht außerhalb eines vordefinierten Volumens befinden. Die Selbstorganisation von Bäumen durch die Konkurrenz um die Ressourcen Licht und Raum stellt eine Erweiterung dieser Methode durch Palubicki et al. [6] dar, die in diesem Bericht noch genauer erläutert wird (vgl. Abschnitt 2.1.1).

Geometrische Methoden wie z.B. die von Weber et al. [7] variieren direkt die geometrischen Größen, wie Astlänge oder -winkel, sodass eine große Anzahl von Parametern für die Erzeugung eines bestimmten Modells angepasst werden muss. Als Alternative kann man Bilder von Bäumen als Eingabe verwenden, was zu sehr guten Ergebnissen führen kann, aber den Nachteil hat, dass die Bilder bereitgestellt werden müssen. Eine dieser Methoden wird von Neubert et al. [8] beschrieben, welche Partikelflussimulationen verwendet und später näher erläutert wird, um ihre Ergebnisse mit der von Xu und Mould vorgestellten Methode zu vergleichen (vgl. Abschnitt 2.1.2).

Um die Benutzerkontrolle über erzeugte Baummodelle zu erhöhen, bieten einige Methoden die Möglichkeit, das Modell mittels Skizzen zu beeinflussen [9, 10, 11, 12]. So wird die Wuchsrichtung des auf L-Systemen beruhenden Verfahrens von Ijiri et al. [9] durch Malen von Strichen

vom Benutzer kontrolliert. Einen komplexen Baum zu modellieren benötigt sehr viel Benutzerinteraktion.

Unter der Annahme, dass die Äste von Bäumen den größtmöglichen Abstand voneinander haben, generieren Okabe et al. [10] aus zweidimensionalen Skizzen dreidimensionale Modelle von Bäumen. Dies ermöglicht es auch ungeübten Benutzern natürlich aussehende Bäume interaktiv und schnell zu gestalten. Chen et al. [11] verwenden probabilistische Optimierung basierend auf Parametern, die aus Modelldatenbanken gewonnenen werden, um 3D-Baummodelle aus Skizzen zu generieren. Die Interaktion der Äste wird durch ein Markov-Random-Field erzeugt, das die Projektion des Modells auf die 2D-Skizze berücksichtigt. Auf ähnlichen stochastischen Optimierungen beruht die Methode von Wither et al. [12], die mit Hilfe von vorhandenem botanischen Wissen von Skizzen der Baumkronenform auf Äste und Blätter des 3D-Baummodells zurückschließt.

Im Gegensatz zu den anderen Methoden ist 3D-Scanning zum Erzeugen von Punktmengen, die das Modell beschreiben, vor allem dafür geeignet Modelle von real existierenden Bäumen zu erstellen. Dies führt zu hoher Qualität, bietet jedoch nicht die Möglichkeit vollkommen neue Bäume zu erschaffen. Bucksch et al. [13] extrahieren ein Baumskelett aus der gescannten Punktmenge, indem sie Cluster von Punkten zusammenfassen und die Punkte räumlich unterteilen. Livny et al. [14] wenden Globaloptimierung an, um mehrere sich überlappende Bäume aus einer Punktwolke zu rekonstruieren. Xu und Mould rekonstruieren Baumskelette aus einer 3D-Scanpunktwolke, indem sie benachbarte Punkte zu einem Graphen verknüpfen und Äste mittels Suche nach dem kürzesten Weg mit einem Wurzelknoten konstruieren [15].

Diese Idee des Pfadsuchens zur Konstruktion von verästelten natürlichen Phänomenen geht auf Xu und Mould zurück, die die Methode für Blitze, Korallen und Bäume untersucht haben [16, 17]. Dabei wird als Graph ein zwei- oder dreidimensionales Gitter verwendet, was dazu führt, dass in den resultierenden Modellen die regelmäßige Struktur in den Ästen wiederzufinden ist. Dies mit einer hohen Auflösung des Gitters zu kompensieren benötigt sehr viel Speicher, weshalb in den Arbeiten von Xu und Mould, die hier genauer vorgestellt werden, als Ansatz ein unregelmäßiges Gitter verwendet wird [1, 2].

2.1 Selbstorganisierende und bildbasierte Baummodelle

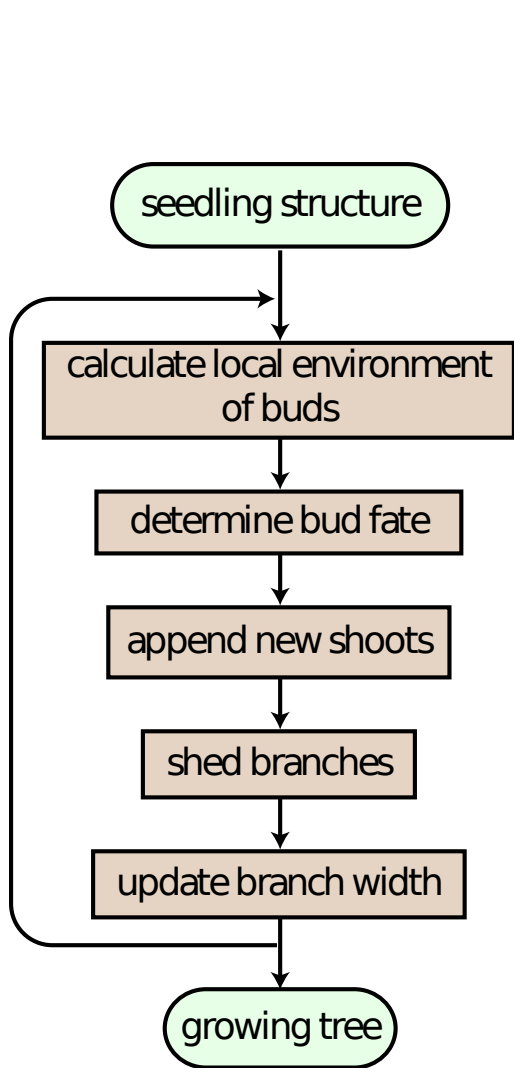
Die bereits erwähnten Methoden von Palubicki et al. [6] und Neubert et al. [8] werden im Folgenden näher ausgeführt, da die mit diesen Methoden erhaltenen Baummodelle mit denen von Xu und Mould [1, 2] in Abschnitt 3.5 verglichen werden.

2.1.1 Selbstorganisierende Baummodelle: „Self-organizing tree models for image synthesis“

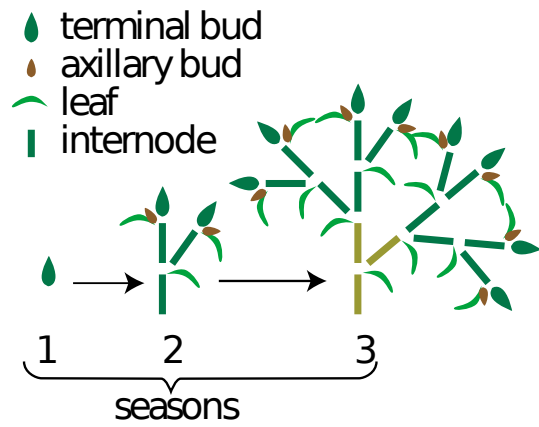
Der Methode von Palubicki et al. liegt die biologische Hypothese zugrunde, dass die Form eines wachsenden Baumes aus einem sich selbst regulierenden Prozess entsteht, der bestimmt wird vom Wettstreit der Knospen und Äste um die Ressourcen Licht und Raum. Die Baumstruktur wird dabei durch Simulation des Entwicklungsprozesses simuliert, wobei der Baum und seine Umgebung sich rückkoppelnd gegenseitig beeinflussen (vgl. Abb. 1(a), 1(b)).

Dabei wird zunächst für jede Knospe bestimmt, ob um sie herum Platz zur Verfügung steht und in welche Richtung ein neuer Ast am besten wachsen könnte. Hierfür stellen Palubicki et al. zwei verschiedene Methoden vor: Zum einen die Raumausnutzung¹ bei der den Knospen gleichmäßig verteilte Markerpunkte aus ihrer Umgebung zugeordnet werden, bzw. die Marker werden der Knospe, die am dichtesten daran ist, zugewiesen (vgl. Abb. 1(c)). Eine Knospe hat Platz zum Wachsen, wenn ihr mindestens ein Marker zugeordnet ist. Die Wachstumsrichtung ergibt sich aus der normierten Summe der Vektoren von der Knospe zu den ihr zugeordneten Markerpunkten.

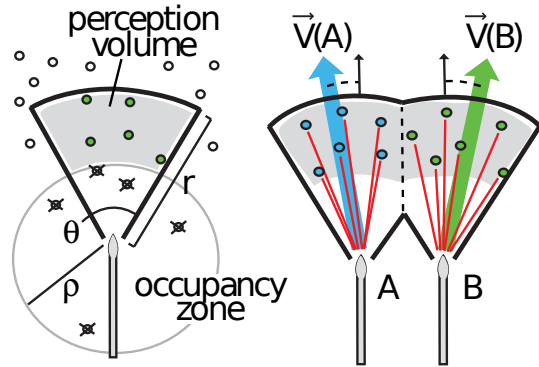
¹„Space Colonization“, [6]



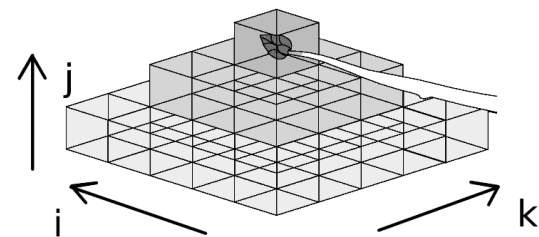
(a) Ablauf der Berechnungen.



(b) Entwicklung des Modells basierend auf L-Systemen. Graphik ist verändert nach [6].



(c) Bestimmung des für das Wachstum zur Verfügung stehenden Platzes und der Wuchsrichtung.



(d) Ausbreitung des Schattens einer Knospe.



(e) Fertige Modelle.

Abbildung 1: Veranschaulichung der Methode nach Palubicki et al. [6]

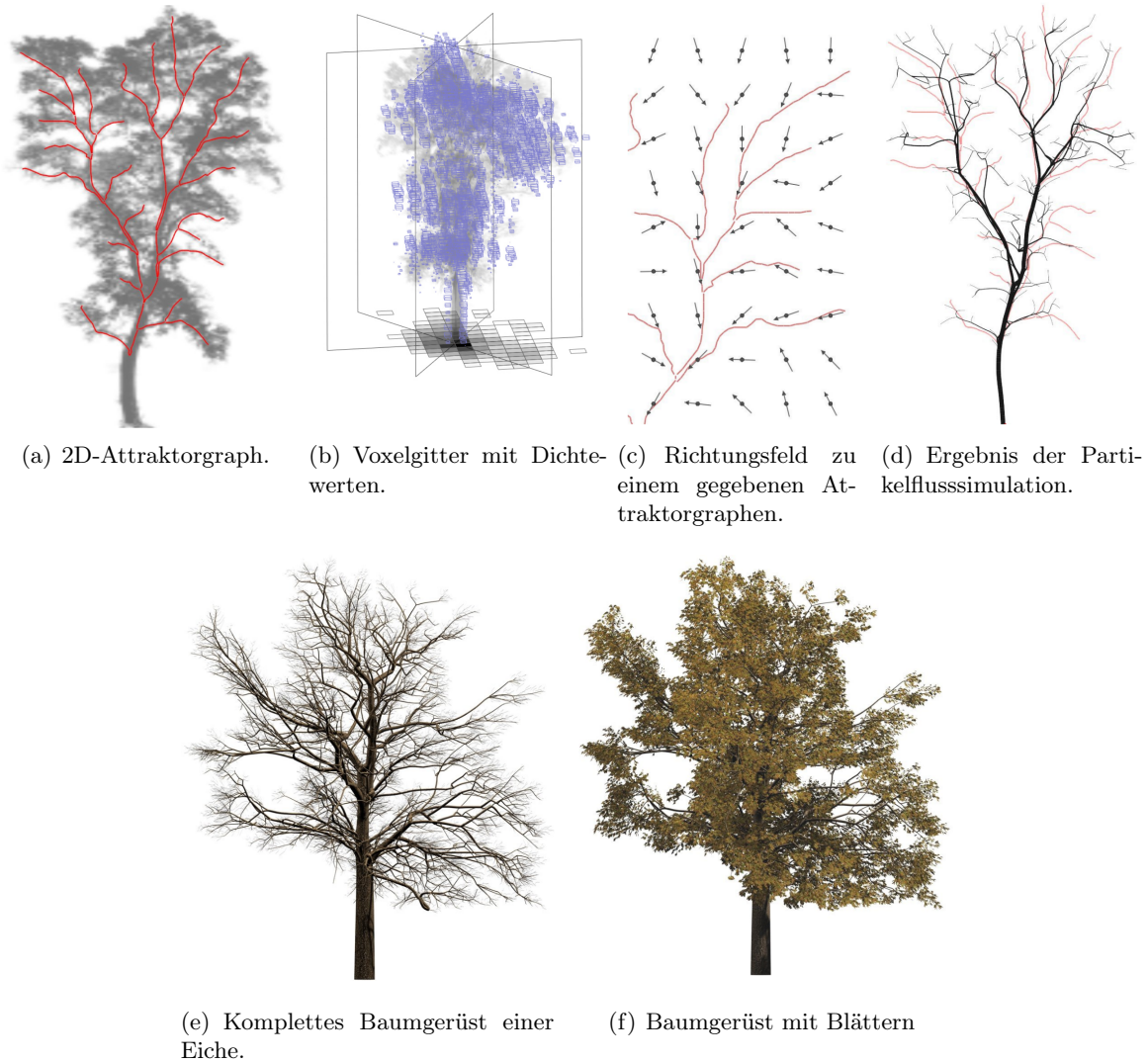


Abbildung 2: Bilder zu einzelnen Schritten der Methode von Neubert et al. [8].

Zum anderen wird die Ausbreitung von Schatten² berücksichtigt. Dabei wird der zur Verfügung stehende Raum in Voxel aufgeteilt, denen jeweils ein Schattenwert zugeordnet wird, der standardmäßig null beträgt. Eine Knospe erzeugt in den Voxeln um sich herum einen pyramidenförmigen Halbschatten, wie in Abbildung 1(d) dargestellt ist. Aus den Schattenwerten der umgebenden Knospen wird berechnet, ob genug Platz zur Verfügung steht, und die Wachstumsrichtung ergibt sich aus dem negativen Schattengradienten. Die so bestimmten Umgebungsparameter der Knospen werden zusammen mit weiterem botanischen Wissen verwendet, um Wachstumsprioritäten für die Knospen zu berechnen.

Um Benutzerkontrolle über die Baummodelle zu gewährleisten kann vor allem die Umgebung interaktiv manipuliert werden. Mit Hilfe von skizzierten Linien können Knospen und Wachstumsrichtungen priorisiert werden und manuelles Setzen von Markerpunkten kann den Raum, der für das Wachstum von neuen Ästen zur Verfügung stehen soll, ausweisen.

2.1.2 Bildbasierte Baummodelle: „Approximate Image-Based Tree-Modeling using Particle Flows“

Die Methode von Neubert et al. verwendet Fotografien als Eingabe um daraus mittels Partikel-flusssimulation 3D-Modelle von Bäumen zu erstellen, die denen auf den Eingabebildern ähnlich sehen. Dabei wird in mehreren Schritten vorgegangen, die im Folgenden kurz aufgelistet und erläutert werden:

1. Pre-Processing: In den Eingabebildern werden zunächst die Bäume vom Hintergrund getrennt. Da Bäume meist Lücken in den Baumkronen besitzen, ist dies kein trivialer Prozess, der nicht komplett automatisch ausgeführt werden kann. Um das zugrundeliegende Astgerüst des Baumes zu rekonstruieren, wird der 2D-Attraktorgraph (vgl. Abb. 2(a)) der Bilder berechnet. Hierfür müssen zuvor Ankerpunkte gesetzt werden, wobei manuelles Setzen die besseren Ergebnisse produziert.
2. Erzeugen des Voxelmodells: Über jedes Eingabebild wird ein 2D-Gitter gelegt, dessen Pixel Dichtewerte zugeordnet bekommen, die eine Schätzung der Baumdicke an der entsprechenden Stelle sind. Hierbei wird davon ausgegangen, dass von dem jeweiligen Baum mindestens zwei Bilder aus jeweils unterschiedlichen Blickrichtungen vorliegen. Die Dichtewerte der Ebenen werden dann in den entsprechenden Winkeln zusammengefügt und zu einem 3D-Gitter ergänzt, dessen Voxel die kombinierten Dichtewerte enthalten (vgl. Abb. 2(b)).
3. Berechnung des Richtungsfeldes: Entlang des Attraktorgraphen einer Ebene wird ein Vektorfeld erzeugt, wie in Abbildung 2(c) zu sehen ist. Die Kombination der Felder für alle Bildebenen wird verwendet um die Teilchen in der folgenden Partikel-flusssimulation zu leiten.
4. Partikel-flusssimulation: Proportional zu den Dichtewerten des Gitters wird eine Anzahl von initialen Partikeln im jeweiligen Voxel zufällig positioniert. Die Bewegung der Partikel wird durch die Kräfte der Nachbarn und durch das Vektorfeld beeinflusst. Als Ergebnis bilden die Spuren der Teilchen das Baumskelett des Modells (vgl. Abb. 2(d)).
5. Erzeugen der Geometrie: Mittels allometrischer Berechnungen wird das 3D-Modell des Baumes erzeugt. Kleine Äste und optional Blätter werden hinzugefügt (vgl. Abb. 2(e), 2(f)).

Die Methode von Neubert et al. bietet auf Grund des starken Einflusses der Eingabebilder nur wenige Möglichkeiten für Benutzer Einfluss auf die Form des resultierenden Modells zu nehmen. Durch die realen Vorbilder hat dieses jedoch ein sehr natürliches Aussehen.

3 Der Algorithmus von Xu und Mould

Der Algorithmus von Xu und Mould [1] zur Erstellung von Baummodellen mittels Pfadsuche in einem unregelmäßigen Gitter lässt sich in vier Teilschritte zerlegen:

1. Erzeugen eines Graphen und Setzen von zufälligen Kantengewichten.
2. Auswählen eines Knotens als Wurzel und weiterer als Endpunkte.
3. Suchen der Pfade mit den geringsten Kosten von den Endpunkten zur Wurzel.
4. Erzeugen von geometrischen Körpern um die Pfadsegmente und Rendern des resultierenden Modells.

²„Shadow Propagation“ [6]

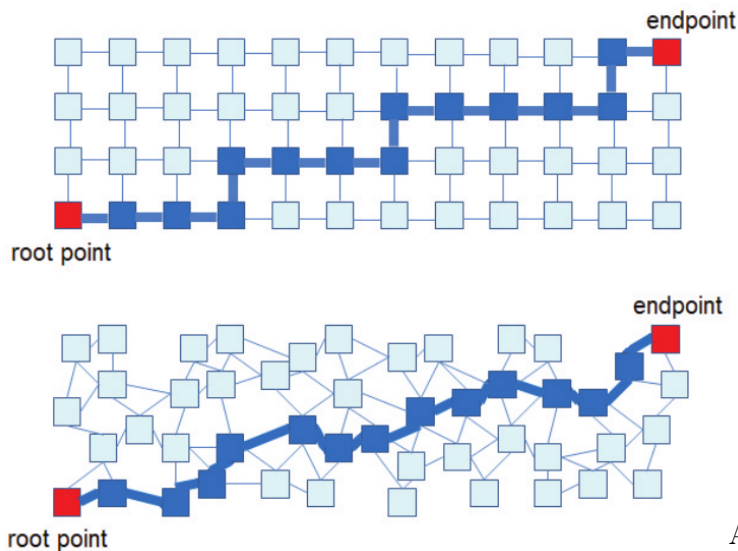


Abbildung 3: Ein Pfad in einem regelmäßigen und in einem unregelmäßigen Gitter[1].

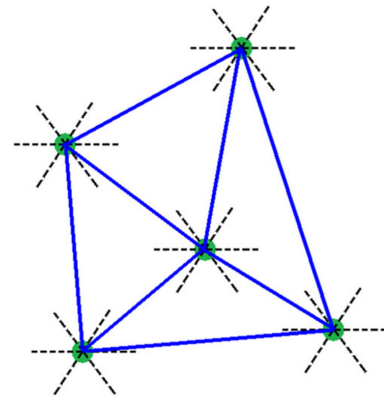


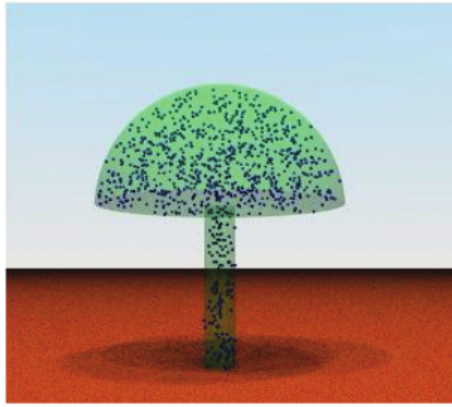
Abbildung 4: Zweidimensionaler Yao-graph mit sechs Sektoren [2].

Bei der Erzeugung des Graphen im ersten Schritt werden Knoten poissonverteilt in einem vorgegebenen Volumen generiert, um ein unregelmäßiges Gitter zu erhalten. Im Vergleich zur Verwendung eines regelmäßigen Gitters, wie sie zuvor von Xu und Mould beschrieben wurde [16], sind hier bei gleicher Auflösung, also gleicher Anzahl an Knoten, keine Überreste rechter Winkel des Gitters im Modell zu sehen (vgl. Abb.3), was den resultierenden Ästen ein natürlicheres Erscheinungsbild verleiht.

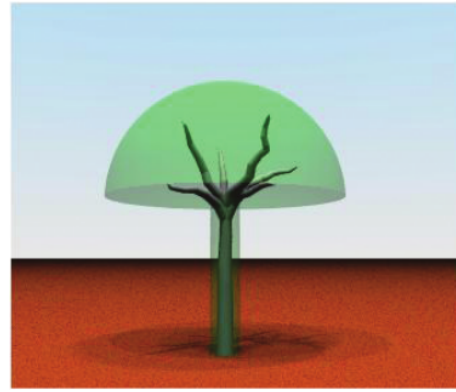
Um zu entscheiden, welche Knoten des Graphen verbunden werden, kann ein maximaler Abstand der Knoten festgelegt werden, sodass nur zwischen Knoten, deren Abstand kleiner ist, Kanten ausgebildet werden. Dies funktioniert für die meisten Graphen, allerdings kann die Wahl eines zu hohen Grenzwertes dazu führen, dass sehr viele Kanten von einem Knoten ausgehen, was speicherintensiv ist. Liegt der Wert zu niedrig, führt dies zu disjunkten Subgraphen. Daher wird von Xu und Mould als weitere Alternative die Verwendung eines Yao-graphen vorgeschlagen [2]. In diesem wird die Umgebung jedes Knotens in Sektoren unterteilt und in jedem Sektor eine Kante zum jeweils nächstliegenden Knoten ausgebildet (vgl. Abb. 4). Im zweidimensionalen Fall erfolgt die Aufteilung mittels Strahlen, im dreidimensionalen mittels Ebenen. Im resultierenden Graphen kann ein Knoten mehr als eine Kante pro Sektor besitzen, da Kanten, die von anderen Knoten ausgehen, hinzukommen können. Die Methode gewährleistet, dass die Anzahl der Kanten limitiert ist, und stellt gleichzeitig sicher, dass in alle möglichen Richtungen Verbindungen zum Rest des Graphen existieren.

Im zweiten Schritt des Algorithmus hat besonders die Wahl der Endpunktpositionen starken Einfluss auf die Form des resultierenden Baummodells. Im hier betrachteten Pseudocode (vgl. Algorithmus 1) werden diese zufällig im gesamten Graphenvolumen gewählt. Alternativen sind, den Bereich, in dem die Endpunkte liegen können, einzugrenzen oder sie manuell zu setzen. Die Pfade mit den optimalen Kosten, die im dritten Schritt für jeden Endknoten erzeugt werden, werden zu einem Subgraphen zusammengefügt. Je weiter dabei ein Pfadsegment des Subgraphen von der Wurzel entfernt liegt, desto dünner ist der Durchmesser des Zylinders, der im vierten Schritt um das Segment gebildet wird.

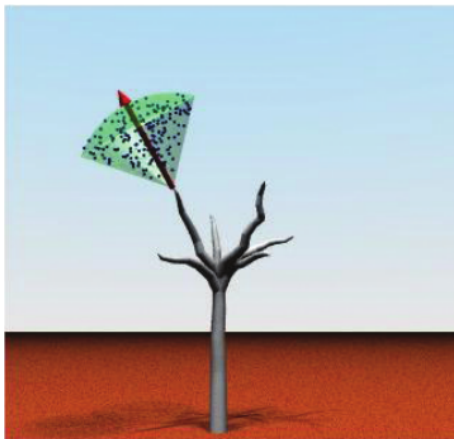
Die Tatsache, dass bei einem Baum vom Stamm Äste und von denen wiederum Zweige ausgehen, zeigt die Möglichkeit seine Struktur rekursiv zu modellieren. Dies wird genutzt, indem die oben beschriebenen Schritte mehrfach angewendet werden, um ausgehend von den Endpunkten des



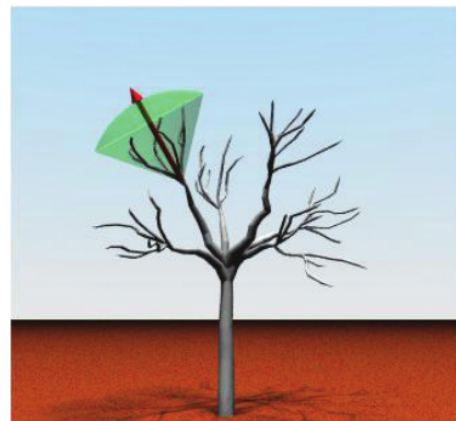
(a) Initiales Graphvolumen aus einer Halbsphäre und einem Zylinder mit unregelmäßig verteilten Knoten.



(b) Modell nach dem initialen Durchlauf des Algorithmus.



(c) Kegel als Subgraphvolumen setzt an einem Endpunkt des Initialgraphs an.



(d) 3D-Modell nach der zweiten Stufe.

Abbildung 5: Illustration des stufenweisen Modellbildungsprozesses [1].

ursprünglichen Graphen Subgraphen hinzuzufügen und den Baum stufenweise aufzubauen. Der Algorithmus von Xu und Mould wurde iterativ implementiert, jedoch als rekursiver Pseudocode vorgestellt, um diese Struktur zu verdeutlichen (vgl. Algorithmus 1).

In der ersten Rekursionsstufe wird das grundlegende Graphvolumen mittels Komposition primitiver geometrischer Formen festgelegt (vgl. Abb. 5(a)) und in diesem der vorläufige Graph mittels des Algorithmus gebildet (vgl. Abb. 5(b)). In den weiteren Rekursionsstufen wird an jedem Endpunkt des bereits bestehenden Graphen ein Subgraph in einem Kegelvolumen angesetzt (vgl. Abb. 5(c) und 5(d)). Je nach gewünschter Detailtiefe kann die Anzahl der Rekursionen beispielsweise 4 oder auch mehr betragen. Mit dem Parameter der Lebensspanne kann die Rekursionstiefe für die verschiedenen Äste variiert werden, dabei erhält ein Endpunkt eine kleinere Lebensspanne als der Elternknoten. Hat die Lebensspanne den Wert null erreicht, wird kein weiterer Subgraph am Endpunkt des Astes mehr erzeugt.

Am Ende dieses Prozesses ergibt sich eine Menge von Kanten, die das Grundgerüst des Baummodells bilden, und um die herum Zylinder platziert werden, um die Geometrie zu erzeugen. Die Dicke w der Zylinder der i -ten Stufe ergibt sich aus der Entfernung des jeweiligen Segments zur Astspitze d' und einem Verdünnungsfaktor ζ : $w = (d')^\zeta / (i+1)$. Größere Werte für ζ verdünnen die Äste schneller. Xu und Mould verwenden typischerweise $\zeta = 0.3$. Zur Darstellung der Baummodelle verwenden Xu und Mould mehrfach die Zylinder gezeichnet als schwarze Flächen auf



Abbildung 6: Durch verschiedene Graphvolumina erhaltene Modelle

weißen Hintergrund, wodurch eine Projektion der Baumstruktur entsteht. Es werden aber auch fotorealistisch gerenderte Modelle erstellt.

3.1 Variation der Parameter

Die Hauptmöglichkeiten das Aussehen des Modells zu beeinflussen, sind die Variation der Form des initialen Graphvolumens, des Subgraphvolumens und der Lebensspanne. Das Graphvolumen hat einen Effekt auf die generelle Form des Baumes. Kontrolliertes Verändern des Subgraphvolumens beeinflusst die Entwicklung des Modells nach der ersten Stufe und ermöglicht geschickteres Eingreifen in das Aussehen des Baums. Die Variation der Lebensspanne beeinflusst die Entwicklung einzelner Äste und bietet die Möglichkeit asymmetrische und unregelmäßige Aspekte hinzuzufügen. Genauere Beschreibungen von Xu und Mould zur Variation der Parameter [1, 2] werden im Folgenden vorgestellt.

3.1.1 Form des initialen Graphvolumens

Das initiale Graphvolumen bestimmt die grundlegende Form des Baummodells. Zwei Methoden das Graphvolumen zu erhalten werden hier beschrieben: Zum einen die benutzerdefinierte Kombination einfacher geometrischer Körper und zum anderen eine Form der Verwendung von Benutzerskizzen.

Bei der ersten Methode werden Formen wie Zylinder, Sphären oder Ellipsoide so angeordnet, dass sich eine Annäherung an die gewünschte Form des Baumes ergibt. Abbildung 6 zeigt drei verschiedene Baummodelle mit ihren zugehörigen initialen Graphvolumina. Die resultierenden Strukturen entsprechen nicht genau den Formvorgaben, was vor allem auf das Hinzufügen der Subgraphen in den weiteren Stufen zurückzuführen ist, aber der Zusammenhang mit dem ursprünglichen Volumen wird deutlich.

Durch geeignete Benutzerdefinition ist es möglich auch spezifischere Vorgaben umzusetzen, so zeigt Abbildung 18 einen Versuch ein Modell zu erzeugen, das eine Fotovorlage imitiert. Dazu wurde der initiale Graph angepasst und die Endpunkte der ersten Stufe wurden manuell gesetzt. Die Verwendung von einfachen geometrischen Formen erzeugt Modelle mit einem gleichmäßigen Aussehen. In Realität besitzen Bäume jedoch Häufungen von Ästen, Lücken in der Baumkrone und unterschiedlich lange Äste. Eine Möglichkeit die Modelle in dieser Richtung zu verbessern, ist, den Bereich für die Auswahl der Endpunkte genauer festzulegen [2], was Einfluss auf die

Globale Parameter:

- Subgraphauflösung k (Anzahl der Knoten)
- Verknüpfungsabstand der Knoten d
- Schrumpfparameter des Subgraphen a
- Subgraphwinkel α

Argumente:

- Graphvolumen V
- Wurzelknoten R
- Verzweigungsfaktor b
- Anzahl der Graphknoten K
- Subgraphradius r
- Lebensspanne des Astes L

Pseudocode *maketree*:

1. Erzeuge den Graphen G für das aktuelle Volumen
 - A. $G \leftarrow \emptyset$
 - B. Finde zufällige Position \vec{p}
 - C. Falls \vec{p} außerhalb von V
verwerfe \vec{p}
 - D. Für alle Knoten $N \in G$:
Falls $|\vec{p} - N.\vec{x}| < d$
verwerfe \vec{p}
 - E. Sonst
erzeuge Knoten m
 $m.\vec{x} = \vec{p}$
 $G \leftarrow \emptyset$
 - F. Solange $|G| < K$ wiederhole B bis E
 - G. Für alle Paare von Knoten $n, m \in G$:
Falls $|n.\vec{x} - m.\vec{x}| < d$
erzeuge eine Kante
 - H. Für alle Kanten
setze zufällige Kosten
2. Wiederhole für b Endpunkte e :
 - A. Setze $e.\vec{x} \leftarrow$ zufällige Position in V
 - B. Für alle Knoten $n \in G$
Falls $|e.\vec{x} - n.\vec{x}| < d$
Erzeuge eine Kante von n zu e $T \leftarrow \emptyset$
3. Erzeuge Pfade zu allen Endpunkten e
 - A. Finde die Sequenz P von Kanten mit den geringsten Kosten von e zu R .
 - B. Für alle Kanten $E \in P \setminus T$:
 $T \leftarrow \{T, E\}$
4. Rekursion für alle Endpunkte:
 - A. Für alle Endpunkte e :
 - B. $\vec{v}_g = (e.\vec{x} - R.\vec{x})/|e.\vec{x} - R.\vec{x}|$
 - C. Definiere V Kegel mit Spitze bei $e.\vec{x}$, mit Radius r , Mittelpunktswinkel α und Mittelachse in Richtung \vec{v}_g
 - D. Falls $L > 0$:
 $T \leftarrow \{T, \text{maketree}(V, e, b, k, a \cdot r, L - 1)\}$
5. Gib T zurück

Algorithmus 1: Parameter und Pseudocode

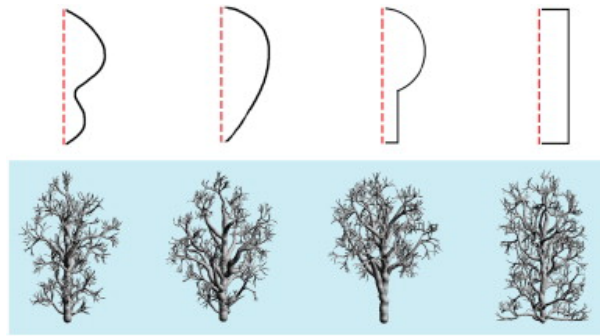


Abbildung 7: Durch Benutzerskizzen beeinflusste Modelle. In der oberen Zeile befinden sich manuell gezeichnete Kurven mit gestrichelter Rotationsachse, darunter die in dem Rotationsvolumen erzeugten Modelle [1].



Abbildung 8: Links: Initialer Graph. Mittig: Fotovorlage. Rechts: Baummodell. [1]

Ausprägung der Astlängenunterschiede hat. Xu und Mould definieren hierzu eine Schale der Dicke d_e innerhalb des Graphvolumens, in der die Endpunkte liegen müssen. Eine schmale Schale, mit kleinem d_e , führt zu einer geringen Astlängenvariation des Modells. Bei größerem d_e ergibt sich eine breitere Schale für die Endpunkte und die resultierenden Astlängen unterscheiden sich stark (vgl. Abb. 9).

Nach vier Iterationsdurchläufen des Algorithmus hat das Modell mit der dickeren Endpunktschale ein asymmetrischeres Aussehen als das mit dem schmalen Volumen für die Endpunkte. Letzteres nutzt durch die Endpunktplatzierung am äußeren Rand das initiale Graphvolumen stärker aus und entspricht damit eher der Formvorgabe.

Für mehr Unregelmäßigkeit in der Baumkrone beschreiben Xu und Mould die Möglichkeit das initiale Graphvolumen mit kleineren Körpern auszufüllen. Als eine mögliche Umsetzung werden hier Punkte mit großem Abstand im Volumen verteilt und um diese Sphären mit festem Radius gebildet. Diese werden mittels Kegel mit dem Stamm oder der Wurzel verbunden, sodass für jeden Punkt in den so entstehenden Körpern eine Verbindung zur Wurzel existiert und dass so disjunkte Subgraphen gebildet werden, die eine Struktur im Modell erzwingen (vgl. Abb. 11). Die Sphären mit den Kegeln füllen auf diese Weise grob das ursprüngliche Graphvolumen aus und ersetzen dieses nun für die Ausführung der ersten Stufe des Algorithmus, wobei die Subgraphen der weiteren Stufen wie zuvor hinzugefügt werden.

Abbildung 12 zeigt, dass die Modelle, die durch die Verfeinerung des initialen Graphvolumens entstanden sind, durch das Erzwingen der disjunkten Subvolumina merklich mehr Unregelmäßigkeiten aufweisen. Auch die Komplexität der Aststruktur ist höher, die Silhouette weicht stärker von der primitiv geometrischen Formvorgabe ab und die Äste sind sichtlich in dichteren Häufungen angeordnet, was das natürliche Aussehen des Baumes verbessert.

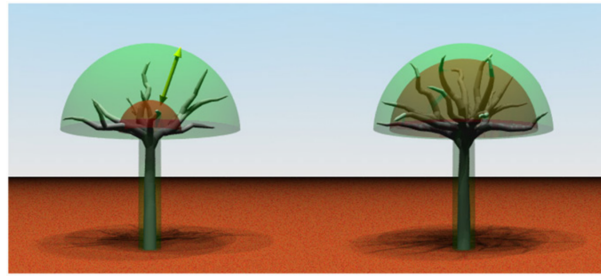


Abbildung 9: Zwei Modelle mit unterschiedlich dicken Schalen für die Endpunktplatzierung. Die Endpunkte werden zwischen der äußeren und der inneren Hemisphäre platziert. Links: Eine dickere Schale erzeugt lange und kurze Äste. Rechts: Eine dünne Schale liefert gleichmäßiger aussehende Äste [2].



Abbildung 10: Modelle aus Abbildung 9 nach der vierten Iteration [2].

3.2 Erzeugen der Subgraphen

Für die Spezifikation des initialen Graphvolumens wird Benutzerinteraktion benötigt. Auch die Subgraphen könnten benutzerdefiniert erzeugt werden, dies wäre jedoch mühsam, sodass die Subgraphvolumina nach dem nachfolgend beschriebenen Schema berechnet werden:

Die Form der Subgraphen ist ein kegelförmiger Ausschnitt einer Sphäre, dessen Spitze die Wurzel des Subgraphen darstellt. Das Volumen hat eine Ausrichtung \vec{v}_g , die z.B. durch den normalisierten Vektor von der Wurzel des Subgraphen zur Wurzel des Graphen der vorherigen Stufe bestimmt ist. Das Volumen kann dann definiert werden als die Menge der Punkte, für die der Vektor von den eigenen Ortskoordinaten zur Wurzel des Subgraphen innerhalb eines Winkels α zum Vektor \vec{v}_g liegt. Der Subgraph wird genau wie der initiale Graph erzeugt, indem das Volumen mit Knoten bestückt wird und von diesen die, die näher als ein Abstand d beieinander liegen, mit Kanten verbunden werden.

Die Größe der Subgraphen ist durch ihren Radius r festgelegt (vgl. Pseudocode 1), der mit zunehmender Iterationsstufe um den Schrumpffaktor a kleiner wird, da normalerweise $a < 1$ gilt. Abbildung 13 zeigt zwei Modelle, die mit vier Iterationsstufen und unterschiedlichen Schrumpfparametern erstellt wurden. Das linke Modell, für das $a = 0.7$ verwendet wurde, hat kurze Zweige am äußeren Rand der Baumkrone, die Zweige und Äste näher zum Stamm sind länger. Der Baum, der mit $a = 1$ erzeugt wurde, hat eine konstante Subgraphgröße, wodurch die Baumkrone eine gleichmäßigere Dichte besitzt.

Um lange gebogene Äste wie bei einer Weide zu erzeugen, werden Segmente mit unterschiedlichen Subgraphorientierungen \vec{v}_g zusammengesetzt. Diese werden mittels einer stetigen Abbildung von \vec{v}_g in den verschiedenen Iterationsstufen berechnet. Als Abbildung kann beispielsweise eine Rotation um die Horizontale verwendet werden, sodass die resultierende Astform eine Kurve ist, die erst steigt und dann nach unten fällt.

Abbildung 14 zeigt einige Beispiele, die mit der beschriebenen Methode nach vier Iterationsstufen

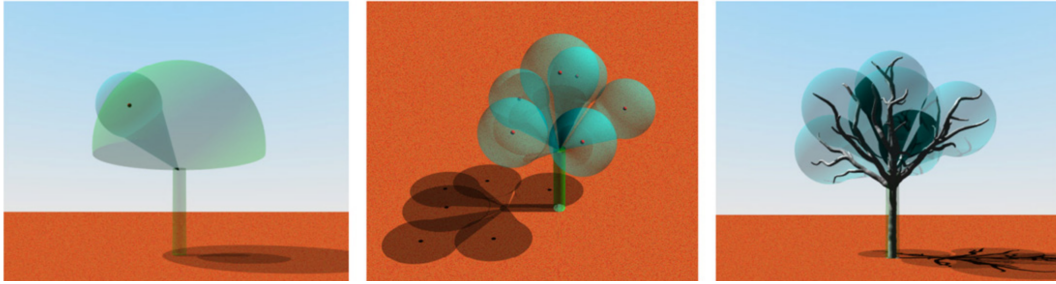


Abbildung 11: Der Prozess, das initiale Graphvolumen mittels Sphären und Kegeln zu strukturieren [2].

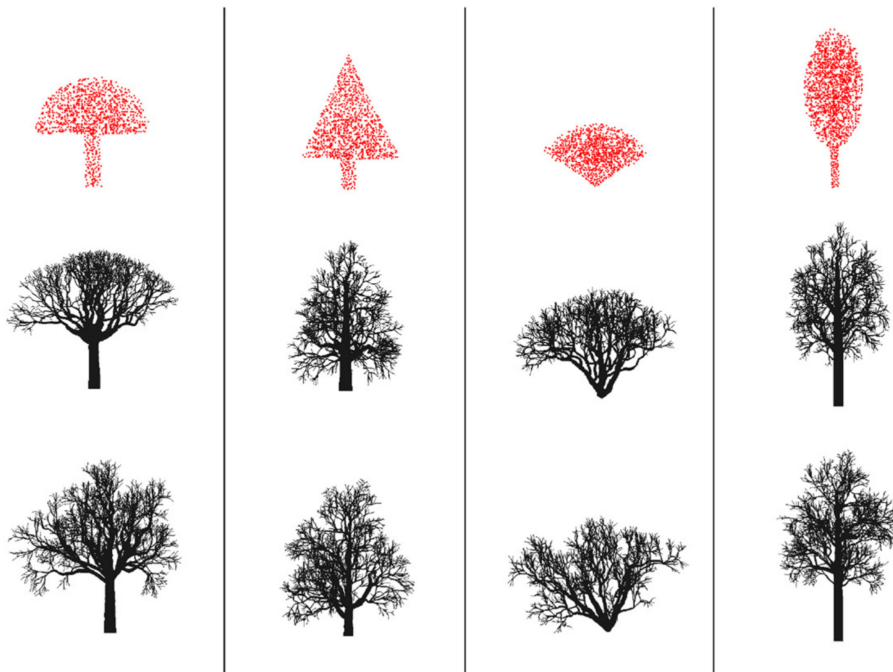


Abbildung 12: Oberste Zeile: Initiale Graphform; Mittlere Zeile: Ohne Verfeinerung des Graphvolumens erzeugte Baummodelle; Unterste Zeile: Mit Verfeinerung des Graphvolumens erzeugte Baummodelle. [2]



Abbildung 13: Das linke Baummodell wurde mit $a = 0.7$ erzeugt, das rechte mit $a = 1$ [1].

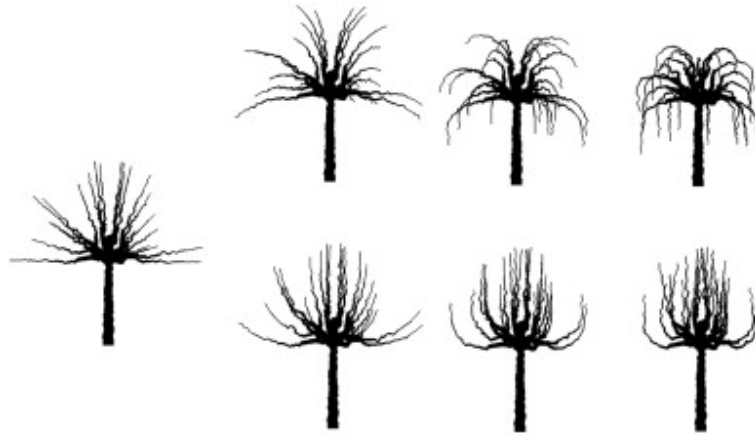


Abbildung 14: Bäume mit gekrümmten Ästen, die durch stückweise Rotation der Subgraphen erzeugt wurden [1].



Abbildung 15: Unregelmäßiger Busch und Baum, erhalten durch Lebensspannenvariation [1].

erzeugt wurden. Xu und Mould haben in ihren Arbeiten vor allem auf diese Methode zurückgegriffen, beschreiben jedoch noch weitere Möglichkeiten die Subgraphen zu verändern, um das resultierende Modell gewünschten Vorgaben anzupassen. So können andere Formen für das Subgraphvolumen oder andere Methoden zur Berechnung der Orientierung verwendet werden. Für letzteres können zufällige oder feste Vektoren dienen oder die Subgraphorientierungen können sich entsprechend Signalen aus der Umwelt anpassen, was in Abschnitt 3.4 näher beschrieben wird.

3.3 Lebensspanne

In der bereits betrachteten Version des Algorithmus 1 wird die Lebensspanne L während jeder Iterationsstufe um eins verringert. Dies führt zu sehr einheitlichen Bäumen mit ähnlich langen Ästen. Durch stärkere Variation der Lebensspanne für die einzelnen Äste können unregelmäßigere Strukturen erzeugt werden.

Xu und Mould geben als Beispiel, dass ein Ast mit einer 30%-igen Wahrscheinlichkeit nach der aktuellen Iterationsstufe endet und dass sich mit einer 70%-igen Wahrscheinlichkeit die Lebensspanne des Astes um einen zufälligen Wert im Intervall $(0, L)$ verringert, wobei L die aktuelle Lebensspanne ist.

Abbildung 15 zeigt zwei Modelle, die mit Lebensspannenvariation erzeugt wurden. Sie erscheinen durch ihre unregelmäßige Struktur natürlicher. Allerdings erzeugt die Methode nicht zuverlässig Modelle mit entsprechender Qualität. Daher nennen Xu und Mould weitere Untersuchungen zur



Abbildung 16: Effekt des Lichtes auf den Verzweigungsfaktor (Endpunktanzahl)[2].



Abbildung 17: Effekt des Lichtes auf die Lebensspanne der Äste [2].

Variation der Lebensspanne als zukünftige Aufgabe.

3.4 Umwelteinflüsse

Das Wachstum von Bäumen wird durch Umweltfaktoren wie Temperatur, Tageslicht und Feuchtigkeit oder auch Windrichtung und Platzbegrenzung beeinflusst. Jeder dieser Faktoren wirkt unabhängig auf den Baum ein, das Zusammenspiel bestimmt jedoch am Ende das Aussehen. Für einzelne Faktoren können die Auswirkungen auf das Modell klar vorausgesagt werden. So bewirken Schatten, dass der Baum in Richtung des zur Verfügung stehenden Lichtes wächst, sodass Äste, die viel Sonnenlicht abbekommen, schneller wachsen und mehr Zweige ausbilden. Anhaltend starker Wind kann das Astwachstum behindern und Hindernisse, wie Mauern, sorgen dafür, dass der Baum in die Richtung wächst, in der mehr Platz zur Verfügung steht.

Xu und Mould beschreiben, dass Umweltfaktoren in ihrem Algorithmus durch die Anpassung von Parametern wie der Anzahl der Endpunkte, der Lebensspanne und der Subgraphorientierung \vec{v}_g berücksichtigt werden können [2]. Dabei beschränken sie sich auf die Umsetzung von Sonnenlichteinflüssen, andere Faktoren können jedoch auf ähnliche Weise eingebaut werden.

Für Äste, deren Ausrichtung mit der Richtung, aus der das Licht kommt, gleich ist, soll die Lebensspanne erhöht und es sollten von ihnen mehr Zweige ausgebildet werden. Unter der Annahme, dass ein Baum n Äste besitzt, an die ein weiterer Subgraph angesetzt werden soll, werden n verschiedene Werte jeweils für die Verringerung der Lebensspanne und der auszubildenden Anzahl der Zweige zufällig aus einem vorbestimmten Bereich ausgewählt. Zusätzlich wird der Winkel zwischen der Lichtrichtung und der Subgraphorientierung berechnet und dann dem Ast mit dem kleinsten Winkel die kleinste Lebensspannenverringerung und die höchste Verzweigungszahl (Anzahl der Endpunkte im Subgraphen) zugeordnet. Dies wird für jeden Ast, geordnet nach ansteigender Winkelgröße, fortgeführt, bis alle Werte verteilt sind. Die Methode sorgt dafür, dass die Äste, die vom Sonnenlicht abgewandt sind, die geringste Lebensspanne haben und die wenigsten Zweige ausbilden.

In Abbildung 16 sind die Auswirkungen der Veränderung der Endpunktanzahl entsprechend des Lichteinflusses in den resultierenden Baummodellen zu erkennen. Der Baum auf der linken Seite, der ohne Berücksichtigung des Lichteinflusses erzeugt wurde, hat eine in alle Richtungen gleichmäßig ausgebreitete Baumkrone. Bei den Modellen auf der rechten Seite variiert die Anzahl

der Zweige entsprechend der Lichtrichtung, sodass der sonnenzugewandte Teil der Baumkrone dichter ist.

Die Auswirkung auf ein Baummodell durch die Veränderung der Lebensspanne entsprechend der Lichtrichtung ist in Abbildung 17 dargestellt. Die Äste, die in Lichtrichtung wachsen, überleben am längsten und werden damit aus den meisten Subgraphen zusammengesetzt.

3.5 Ergebnisse und Evaluation

Im Folgenden werden einige mit dem von Xu und Mould beschriebenen Algorithmus erstellte Ergebnisse vorgestellt und bewertet. Dazu werden die erstellten Baummodelle mit Fotos und Modellen der oben vorgestellten Methoden von Neubert et al. und Palubicki et al. verglichen.

Abbildung 18 vergleicht Fotos mit ihnen nachempfundenen Baummodellen. Diese haben eine ähnliche Struktur wie die fotografierten Bäume, da sie eine ähnliche Baumkrone haben mit wenigen dicken Hauptästen und einem großen Volumen, gefüllt mit Zweigen, zwischen denen sich unregelmäßig verteilte, natürlich aussehende Lücken befinden. Insgesamt ist es schwer zu entscheiden, welche der Bilder reale Bäume und welche generierte Modelle zeigen, sodass die Methode von Xu und Mould als effektiv bezeichnet werden kann.

Der Algorithmus beinhaltet mehrere zufällige Elemente, wie das Setzen der Kantengewichte und der Endpunkte, wodurch es möglich ist, mit den gleichen Parametern ähnlich aussehende, aber nicht identische Modelle zu erhalten. Abb. 19 zeigt drei Baummodelle, die mit einem initialen Graphvolumen aus einer Halbkugel und einem Zylinder erzeugt wurden. Während die allgemeinen Charakteristika der Modelle gleich sind, haben sie doch eine leicht unterschiedliche Struktur. Diese Eigenschaft des Algorithmus ist hilfreich, falls man schnell eine Gruppe von Bäumen der gleichen Sorte erstellen möchte.

Abbildung 20 zeigt Modelle erzeugt aus Graphen, deren Knoten mit den zwei unterschiedlichen Methoden verbunden wurden. Die Modelle aus dem Yaographen und dem mittels geeigneten Grenzwert erstellten Graphen sind gleichwertig, wobei der Yaograph im Vergleich nur etwa die Hälfte der Kanten besitzt. Bei der Wahl eines kleinen Grenzwertes, der zur gleichen Anzahl an Kanten wie beim Yaograph führt, wie in Abbildung 20 in der untersten Zeile, können Lücken im Graphen entstehen und die Endpunkte können zum Teil nicht mit der Wurzel verbunden werden. Als Ergebnis ist zu erkennen, dass der Yaograph gute Ergebnisse bei kleiner Kantenzahl erzielt. Durch Variation der initialen Graphform und der anderen Parameter können viele verschiedene Baumvarianten erzeugt werden (vgl. Abb. 21). Um die verschiedenen Bäume zu erzeugen, wurde vor allem das initiale Graphvolumen verändert, aber auch manuell definierte Subgraphvolumina sind zum Einsatz gekommen. Eine detaillierte Liste der verwendeten Parameter findet sich in den Arbeiten von Xu und Mould [1, 2].

Mit dem beschriebenen Algorithmus können nicht nur Baumkronen, sondern auch Wurzeln generiert werden. Die Strukturen in Abbildung 22 haben Wurzeln, die in einem halbkugelförmigen Volumen erzeugt wurden. Die Hauptwurzel wird durch einen Pfad vom Wurzelpunkt zu einem einzelnen Endpunkt dargestellt. Die Seitenwurzeln werden gebildet, indem um die Hauptwurzel herum zufällig Endpunkte verteilt werden.

Im Vergleich zu anderen Modellierungsmethoden für Bäume stellen Xu und Mould für ihre Methode als Hauptvorteil heraus, dass sie unregelmäßige Äste erzeugen und dabei, anders als bei geometrischen Methoden, nur wenige Parameter brauchen. Die unregelmäßigen Pfade werden durch die Pfadsuche in einem Graphen mit zufällig gewichteten Kanten erzeugt. Daher werden global kontrollierbare unregelmäßige Strukturen durch den Algorithmus ohne weiteren Aufwand generiert.

In den Abbildungen 23 und 24 werden Baummodelle der hier vorgestellten Methode mit Modellen von Neubert et al. [8] und Palubicki et al. [6] verglichen. Da keine objektiven Kriterien für die Qualität eines Baumes existieren, werden die Modelle anhand ihres allgemeinen Aussehens beurteilt.

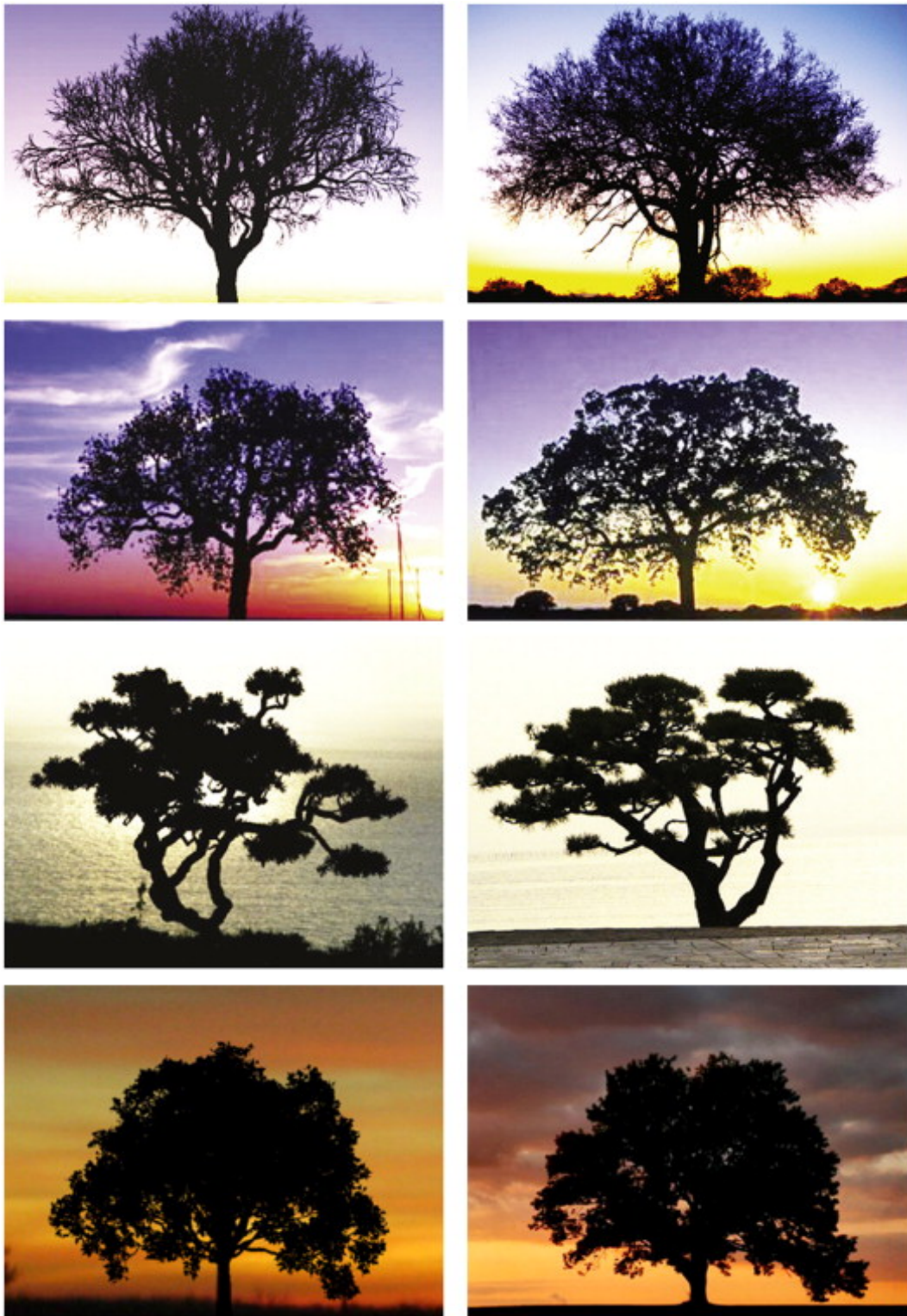


Abbildung 18: Links: Baummodelle von Xu und Mould, Rechts: Fotografien realer Bäume [2].



Abbildung 19: Drei Bäume des gleichen Typs [1].

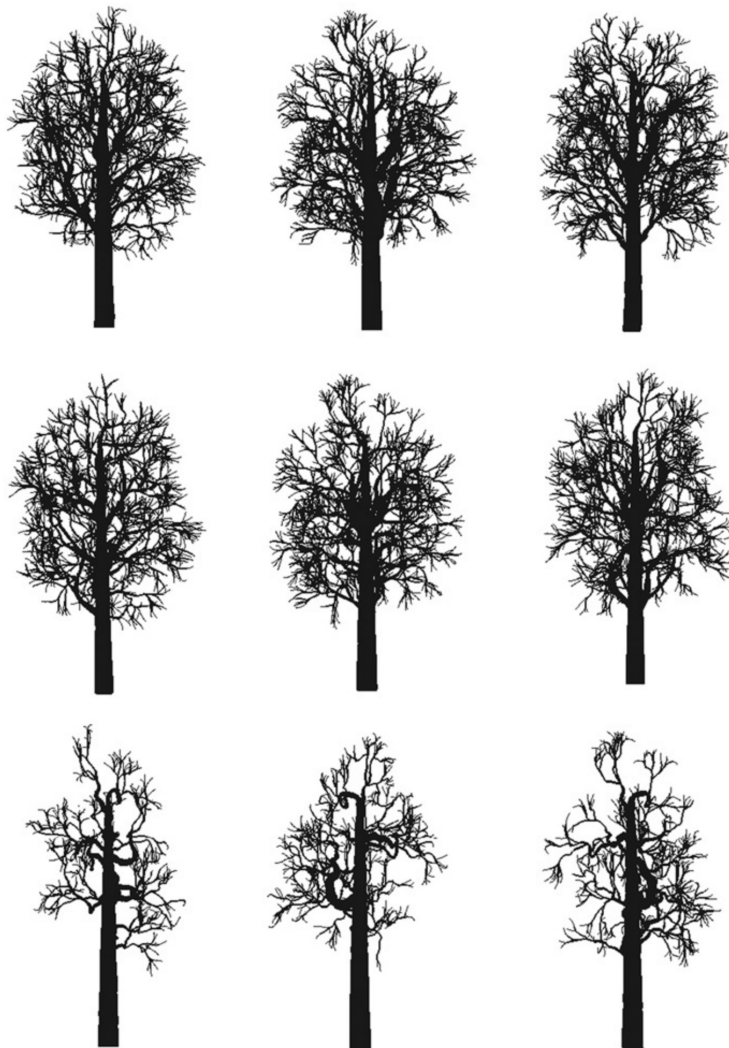


Abbildung 20: Oberste Zeile: Bäume erzeugt aus einem Yaographen; Mittlere Zeile: erzeugt aus einem Graphen, dessen Knoten entsprechend eines geeigneten Abstandsgrenzwertes verbunden wurden; Untere Zeile: Bäume erzeugt aus einem Graphen, dessen Knoten mittels eines kleineren Abstandsgrenzwertes verbunden wurden [2].

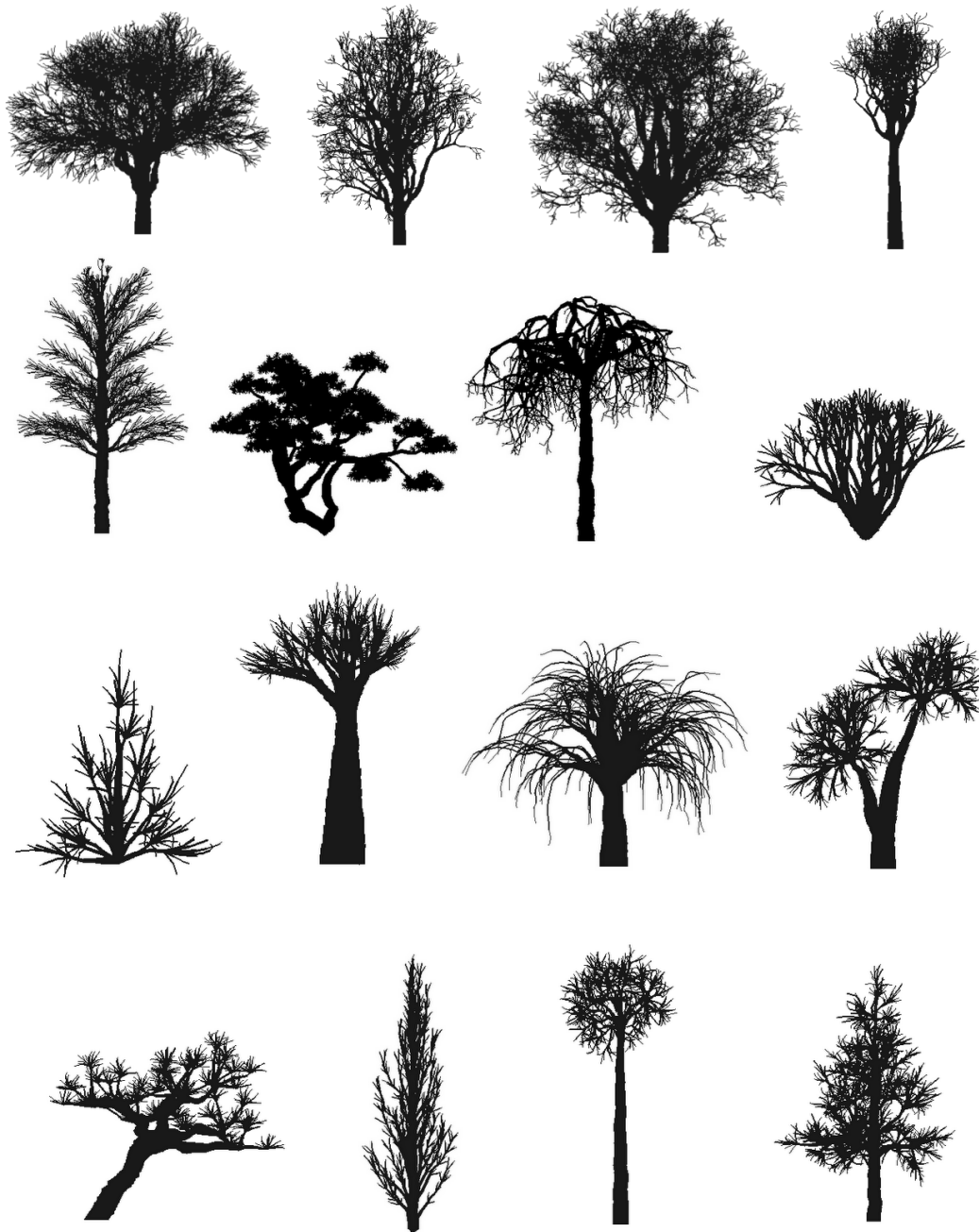


Abbildung 21: Verschiedene Baumtypen. [1]

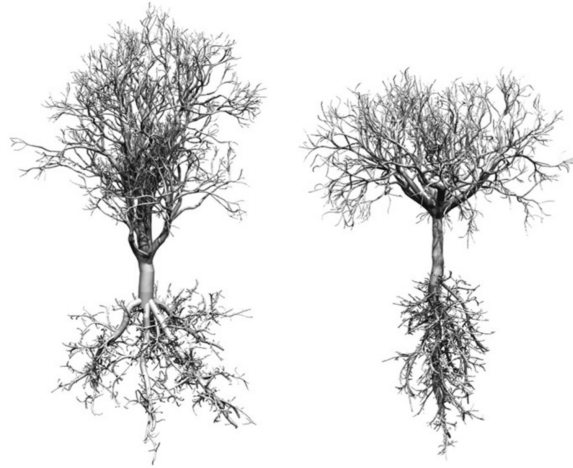


Abbildung 22: Zwei Bäume mit unterschiedlichen Wurzelsystemen [1].



Abbildung 23: Links: Ein selbstorganisierendes Baummodell von Palubicki et al. [6]; Rechts: Ein Baummodell von Xu und Mould [1].



Abbildung 24: Links: Modell von Neubert et al [8].; Rechts: Modell von Xu und Mould [1].

Das Modell von Neubert et al. (vgl. Abb. 24) wurde mittels einer Partikelflussimulation und unter Verwendung einer Fotovorlage erstellt. Beide Modelle der Abbildung haben einen realistischen visuellen Effekt. Meiner Meinung nach ist bei dem Modell von Xu und Mould jedoch die strikte hierarchische Struktur des Algorithmus deutlich zu erkennen, die jedoch von der Natur nicht unbedingt eingehalten wird, was in dem Modell von Neubert et al. durch die Fotovorlage kompensiert wird. Xu und Mould weisen jedoch darauf hin, dass die Verwendung einer solchen einen zusätzlichen Arbeitsschritt darstellt, der gleichzeitig die Möglichkeiten der Benutzer, Einfluss auf das entstehende Modell zu nehmen, auf vorhandene Vorlagen einschränkt.

Abbildung 23 zeigt jeweils ein Baummodell von Palubicki et al. und von Xu und Mould. Beide haben eine ähnliche Form und Aststruktur. In einer 2D-Projektion zeigt sich, dass das mittels selbstorganisierendem Algorithmus erzeugte Modell eine gleichmäßigere Verteilung der Lücken in der Baumkrone besitzt. Je nachdem, welchen Zweck das Modell erfüllen soll, kann dies von Vorteil oder Nachteil sein. Unregelmäßige Bäume mit krummen Ästen sind z.B. dann von Interesse, wenn ein gruseliger Wald dargestellt werden soll.

Die Modelle, die durch Pfadsuche erzeugt werden, sind in ihrer Detailtiefe durch die Anzahl der Knoten im Graphen limitiert, sodass der Algorithmus sehr speicherintensiv ist. Eine Kontrolle über das Ergebnis des Algorithmus wird durch das Setzen der Endpunkte ermöglicht. Sie ist jedoch dadurch eingeschränkt, dass die Pfade das eigentliche Aussehen beeinflussen und diese einen zufälligen Aspekt beinhalten, was, wie bereits erwähnt, seine Vorteile hat.

Mit einer CPU von 3 GHz und RAM von 3 GB konnten Xu und Mould die kleineren Modellen innerhalb von wenigen Sekunden berechnen. Das größte Modell (vgl. Abb. 18, oben links) mit 30000 Endpunkten in einem Graphen mit ungefähr 300000 Knoten benötigte etwa 30 s.

4 Zusammenfassung und Ausblick

Es wurde gezeigt, dass prozedurale Baummodellierung mittels Pfadsuche in Graphen dazu in der Lage ist, komplexe und realistische Bäume zu erzeugen. Das allgemeine Aussehen wird vor allem durch die Form des initialen Graphen und teilweise durch die Subgraphvolumina, die in den weiteren Stufen hinzugefügt werden, beeinflusst. Detailliertere Strukturen werden durch die Verfeinerung des Graphvolumens erreicht. Durch Variation der Parameter existiert die Möglichkeit der Einflussnahme auf die Modelle, was für die meisten Anwendungen benötigt wird. Dadurch können viele verschiedene Modelltypen erstellt werden. Da der Algorithmus zufällige Kantengewichte und Endpunktwahl beinhaltet, lassen sich viele unterschiedliche aber ähnliche Modelle mit den gleichen Parametereinstellungen erzeugen.

Als Hauptbestandteil der zukünftigen Arbeit an der Methode benennen Xu und Mould den Ausbau der Benutzerkontrolle über die resultierenden Modelle. Dafür sollen interaktive Werkzeuge untersucht werden, die Benutzern das Skizzieren von Endpunkten und Markieren von Bereichen mit höheren oder niedrigeren Kantengewichten ermöglichen. Gleichzeitig soll die Einflussnahme auf den initialen Graphen durch Zeichnungen ausgebaut werden. Auch sollen Untersuchungen zu weiteren Aspekten wie Blättern, Früchten und Blüten erfolgen.

4.1 Diskussion

Xu und Mould haben einen einfachen Algorithmus zum Generieren von Baummodellen vorgestellt und seine Vorteile klar herausgearbeitet. Im Vergleich zur vorgestellten Methode von Palubicki et al., sehe ich noch einen klaren Nachteil darin, dass sie zwar die Struktur fertiger Bäume für die Umsetzung des Algorithmus genau analysiert und berücksichtigt haben, jedoch den Aspekt des Wachstumsprozesses nicht betrachten. Auch machen sie nur wenige Angaben darüber, wie zuverlässig der Algorithmus Modelle mit der präsentierten Qualität erzeugt. Durch die Randomisierung der Kantengewichte sind auch ungünstige Konstellationen denkbar, die zu fast geschlossenen Kreisen innerhalb der berechneten Pfade führen könnten, was kein natürli-

ches Aussehen darstellt. Hier wäre interessant, ob dem mit Kontrolle des Abstands zwischen dem größtmöglichen und kleinstmöglichen Kantengewicht entgegengewirkt werden könnte. Die Größe dieses Abstands stellt die maximale Anzahl der Kanten dar, die einen Pfad bilden, der bei der Suche nach den optimalen Kosten der direkten Kante zwischen zwei Knoten vorgezogen werden könnte. Dadurch könnte daher die Größe eines auftretenden fast geschlossenen Kreises begrenzt werden.

Das Grundprinzip des von Xu und Mould beschriebenen Algorithmus baut auf der bekannten und simplen Strategie der optimalen Pfadsuche auf und ist auch ohne tiefes mathematisches oder informatisches Vorwissen zu verstehen. Damit eignet sich der Algorithmus als anschauliches und mit seinen Beispielbildern optisch attraktives Beispiel für die Anwendung von Pfadsuchalgorithmen im Informatikunterricht der Sekundarstufe II am Gymnasium. Als angehende Informatiklehrerin kann ich mir daher vorstellen, den Algorithmus Schülerinnen und Schülern im Unterricht später vorzustellen.

Literatur

- [1] Ling Xu and David Mould. Synthetic tree models from iterated discrete graphs. In *Proceedings of the 2012 Graphics Interface Conference, GI '12*, pages 149–156. Canadian Information Processing Society, 2012.
- [2] Ling Xu and David Mould. A procedural method for irregular tree models. *Computers and Graphics*, 36(8):1036 – 1047, 2012.
- [3] James Scott Hanan. *Parametric L-systems and their application to the modelling and visualization of plants*. PhD thesis, University of Regina, 1992.
- [4] Jerry O. Talton, Yu Lou, Steve Lesser, Jared Duke, Radomír Měch, and Vladlen Koltun. Metropolis procedural modeling. *ACM Trans. Graph.*, 30(2):11:1–11:14, 2011.
- [5] Adam Runions, Brendan Lane, and Przemyslaw Prusinkiewicz. Modeling trees with a space colonization algorithm. In *Eurographics Workshop on Natural Phenomena*, volume 2. The Eurographics Association, 2007.
- [6] Wojciech Palubicki, Kipp Horel, Steven Longay, Adam Runions, Brendan Lane, Radomír Měch, and Przemyslaw Prusinkiewicz. Self-organizing tree models for image synthesis. *ACM Trans. Graph.*, 28(3):58:1–58:10, 2009.
- [7] Jason Weber and Joseph Penn. Creation and rendering of realistic trees. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH '95*, pages 119–128. ACM, 1995.
- [8] Boris Neubert, Thomas Franken, and Oliver Deussen. Approximate image-based tree-modeling using particle flows. *ACM Trans. Graph.*, 26(3), 2007.
- [9] Takashi Ijiri, Shigeru Owada, and Takeo Igarashi. The sketch l-system: Global control of tree modeling using free-form strokes. In *In Smart Graphics*, pages 138–146, 2006.
- [10] Makoto Okabe, Shigeru Owada, and Takeo Igarash. Interactive design of botanical trees using freehand sketches and example-based editing. *Computer Graphics Forum*, 24(3):487–496, 2005.
- [11] Xuejin Chen, Boris Neubert, Ying-Qing Xu, Oliver Deussen, and Sing Bing Kang. Sketch-based tree modeling using Markov random field. *ACM Trans. Graph.*, 27(5):109:1–109:9, 2008.
- [12] J. Wither, F. Boudon, M.-P. Cani, and C. Godin. Structure from silhouettes: a new paradigm for fast sketch-based design of trees. *Computer Graphics Forum*, 28(2):541–550, 2009.
- [13] Alexander Bucksch, Roderik Lindenbergh, Massimo Menenti, and Muhammad Z. Rahman. Skeleton-based botanic tree diameter estimation from dense lidar data. pages 746007–746007–11, 2009.
- [14] Yotam Livny, Feilong Yan, Matt Olson, Baoquan Chen, Hao Zhang, and Jihad El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.*, 29(6):151:1–151:8, 2010.
- [15] Hui Xu, Nathan Gossett, and Baoquan Chen. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.*, 26(4), 2007.

-
- [16] Ling Xu and David Mould. Constructive path planning for natural phenomena modeling. In Dimitri Plemenos and Georgios Miaoulis, editors, *Artificial Intelligence Techniques for Computer Graphics*, volume 159 of *Studies in Computational Intelligence*, pages 83–102. Springer Berlin Heidelberg, 2008.
- [17] Ling Xu and David Mould. Modeling dendritic shapes using path planning. In *GARPP*, pages 29–36. Citeseer, 2007.